IEEE Xplore RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results BROWSE SEARCH IEEE GUID

Results for "(((slicing and optimiz* memory management)<in>metadat (pyr >= 1995 <and&..."
Your search matched 0 documents.
A maximum of 100 results are displayed, 25 to a page, sorted by Relevanc Descending order.

» Search Options

View Session History
New Search

Modify Search

(((slicing and optimiz* memory management)<in>m

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

» Key

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

Indexed by Inspec

IEEE Xplore RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results BROWSE SEARCH IEEE GUID

Results for "(((slicing unused objects)<in>metadata)) <and> (pyr >= 19 pyr <= 200..."
Your search matched 0 documents.
A maximum of 100 results are displayed, 25 to a page, sorted by Relevanc Descending order.

» Search Options

View Session History
New Search

Modify Search

(((slicing unused objects)<in>metadata)) <and> (pyr

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

Indexed by Inspec

IEEE Xplore® RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results

BROWSE SEARCH IEEE GUID

Results for "(((slicing class memory)<in>metadata)) <and> (pyr >= 199 <= 2002..."
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevanc Descending** order.

» Search Options

View Session History
New Search

Modify Search

(((slicing class memory)<in>metadata)) <and> (pyr

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

Indexed by Inspec®

IEEE Xplore RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Results

BROWSE SEARCH IEEE GUID

Results for "(((compact*<in>metadata) <and> (slicer<in>metadata) (memory<...

Your search matched **0** documents.

A maximum of **100** results are displayed, **25** to a page, sorted by **Relevanc Descending** order.

» Search Options

View Session History

New Search

Modify Search

(((compact*<in>metadata) <and> (slicer<in>meta

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL IEEE Journal or Magazine

IEE JNL IEE Journal or Magazine

IEEE CNF IEEE Conference Proceeding

IEE CNF IEE Conference Proceeding

IEEE STD IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer assistance revising your search.

Indexed by Inspec

Published since January 1995 and Published before September 2002
Terms used **slicing class hierarchies**

Sort results by relevance
Display results expanded form
Save results to a Binder
Search Tips
Open results in a new window
Try an Advanc
Try this search

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 ne
Best 200 shown R

1 Slicing class hierarchies in C++
Frank Tip, Jong-Deok Choi, John Field, G. Ramalingam
October 1996 **ACM SIGPLAN Notices , Proceedings of the 11th ACM S conference on Object-oriented programming, systems, la applications OOPSLA '96**, Volume 31 Issue 10
Publisher: ACM Press
Full text available: pdf(2.08 MB) Additional Information: full citation, abst citings, index ter

This paper describes an algorithm for *slicing* class hierarchies in C++ pr C++ class hierarchy (a collection of C++ classes and inheritance relation a program *P* that uses the hierarchy, the algorithm eliminates from the hi members, member functions, classes, and inheritance relations that are u ensuring that the semantics of *P* is maintained.Class slicing is especially program *P* is generated ...

2 Class hierarchy specialization
Frank Tip, Peter F. Sweeney
October 1997 **ACM SIGPLAN Notices , Proceedings of the 12th ACM S conference on Object-oriented programming, systems, la applications OOPSLA '97**, Volume 32 Issue 10

Publisher: ACM Press

Full text available: pdf(2.29 MB) Additional Information: full citation, abst citings, index ter

Class libraries are generally designed with an emphasis on versatility an Applications that use a library typically exercise only part of the library's result, objects created by the application may contain unused members. algorithm that *specializes* a class hierarchy with respect to its usage in a the algorithm analyzes the member access patterns for *P*'s variables, and classes for variable ...

3 Reengineering class hierarchies using concept analysis

Gregor Snelting, Frank Tip

November 1998 **ACM SIGSOFT Software Engineering Notes , Proceed ACM SIGSOFT international symposium on Foundati engineering SIGSOFT '98/FSE-6**, Volume 23 Issue 6

Publisher: ACM Press

Full text available: pdf(1.31 MB) Additional Information: full citation, abst citings, index ter

The design of a class hierarchy may be imperfect. For example, a class C member *m* not accessed in any *C*-instance, an indication that *m* could be moved into a derived class. Furthermore, different subsets of *C*'s membe from different *C*-instances, indicating that it might be appropriate to spli classes. We present a framework for detecting and remediating such desi is ba ...

4 Understanding class hierarchies using concept analysis

Gregor Snelting, Frank Tip

May 2000 **ACM Transactions on Programming Languages and System** Volume 22 Issue 3

Publisher: ACM Press

Full text available: pdf(433.91 KB) Additional Information: full citation, abst citings, index ter

A new method is presented for analyzing and reengineering class hierarc approach, a class hierarchy is processed along with a set of applications fine-grained analysis of the access and subtype relationships between ob class members is performed. The result of this analysis is again a class h

guaranteed to be behaviorally equivalent to the original hierarchy, but in only contains the members that are req ...

Keywords: class hierarchy reengineering, concept analysis

5 Inter-class def-use analysis with partial class representations
Amie L. Souter, Lori L. Pollock, Dixie Hisley
September 1999 **ACM SIGSOFT Software Engineering Notes , Proceed ACM SIGPLAN-SIGSOFT workshop on Program ana tools and engineering PASTE '99**, Volume 24 Issue 5
Publisher: ACM Press
Full text available: pdf(1.27 MB) Additional Information: full citation, abst citings, index ter

Object-oriented program design promotes the reuse of code not only thro polymorphism, but also through building server classes which can be use client classes. Research on static analysis of object-oriented software has addressing the new features of classes, inheritance, polymorphism, and d This paper demonstrates how exploiting the nature of object-oriented des enable development of scalable static analyses. We p ...

6 On the syllogistic structure of object-oriented programming
Derek Rayside, Kostas Kontogiannis
July 2001 **Proceedings of the 23rd International Conference on Softwa**
Publisher: IEEE Computer Society
Full text available: pdf(138.07 KB) Publisher Site Additional Information: full citation, abst citings, index ter

Recent works by Sowa and by Rayside & Campbell demonstrate that the connection between object-oriented programming and the logical forma syllogism, first set down by Aristotle in the Prior Analytics. *In this paper understanding of polymorphic method invocations in terms of the syllogi understanding to the design of a novel editor for object-oriented prograi able to display a polymorphic call graph, which ...*

7 A study of dead data members in C++ applications
Peter F. Sweeney, Frank Tip
May 1998 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLA on Programming language design and implementation PLD** Issue 5
Publisher: ACM Press
Full text available: pdf(1.05 MB) Additional Information: full citation, abst citings, index ter

Object-oriented applications may contain data members that can be remc application without affecting program behavior. Such "dead" data memb to unused functionality in class libraries, or due to the programmer losin usage as the application changes over time. We present a simple and effi detecting dead data members in C++ applications. This algorithm has be using a prototype version of the IBM VisualAge C++ compiler, ...

8 A member lookup algorithm for C++
G. Ramalingam, Harini Srinivasan
May 1997 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLA on Programming language design and implementation PLD** Issue 5
Publisher: ACM Press
Full text available: pdf(1.53 MB) Additional Information: full citation, abst citings, index ter

The member lookup problem in C++ is the problem of resolving a speci in the context of a specified class. Member lookup in C++ is complicate virtual inheritance and multiple inheritance. In this paper, we present an for member lookup in C++. We also present a formalism for the multiple mechanism of C++, which we use as the basis for deriving our algorithm may also be of use as a formal basis for deriving other C++ ...

9 Technical papers: software maintenance: Concern graphs: finding and desc using structural program dependencies
Martin P. Robillard, Gail C. Murphy
May 2002 **Proceedings of the 24th International Conference on Softwa**
Publisher: ACM Press
Full text available: pdf(1.35 Additional Information: full citation, abst

MB) citings, index ter

Many maintenance tasks address concerns, or features, that are not well source code comprising a system. Existing approaches available to help locate and manage scattered concerns use a representation based on lines complicating the analysis of the concerns. In this paper, we introduce the representation that abstracts the implementation details of a concern and relationships between different parts of the co ...

10 Fast static analysis of C++ virtual function calls

David F. Bacon, Peter F. Sweeney

October 1996 **ACM SIGPLAN Notices , Proceedings of the 11th ACM conference on Object-oriented programming, systems, la applications OOPSLA '96**, Volume 31 Issue 10

Publisher: ACM Press

Full text available: pdf(2.10 MB) Additional Information: full citation, abst citings, index ter

Virtual functions make code easier for programmers to reuse but also ma compilers to analyze. We investigate the ability of three static analysis al improve C++ programs by resolving virtual function calls, thereby reduc size and reducing program complexity so as to improve both human and understanding and analysis. In measurements of seven programs of signi 20000 lines of code each) we found that on average the mo ...

11 Practical experience with an application extractor for Java

Frank Tip, Chris Laffra, Peter F. Sweeney, David Streeter

October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM conference on Object-oriented programming, systems, la applications OOPSLA '99**, Volume 34 Issue 10

Publisher: ACM Press

Full text available: pdf(2.31 MB) Additional Information: full citation, abst citings, index ter

Java programs are routinely transmitted over low-bandwidth network co compressed class file archives (i.e., zip files and jar files). Since archive proportional to download time, it is desirable for applications to be as sm paper is concerned with the use of program transformations such as remo methods and fields, inlining of method calls, and simplification of the cl

reducing application size. Such "extract ...

12 Beyond traditional program slicing
Anthony M. Sloane, Jason Holdsworth
May 1996 **ACM SIGSOFT Software Engineering Notes , Proceedings c SIGSOFT international symposium on Software testing and '96,** Volume 21 Issue 3
Publisher: ACM Press
Full text available: pdf(726.83 KB) Additional Information: full citation, abst citings, index ter

Traditional program slices are based on variables and statements. Slices statements that potentially affect (or are affected by) the value of a partic given statement. Two assumptions are implicit in this definition: 1) that statements are concepts of the programming language in which the progr 2) that slices consist solely of statements.Generalised slicing is an extens slicing where variables are replaced by arbitrar ...

13 A schema for interprocedural modification side-effect analysis with pointe
Barbara G. Ryder, William A. Landi, Philip A. Stocks, Sean Zhang, Rita A
March 2001 **ACM Transactions on Programming Languages and Syste** Volume 23 Issue 2
Publisher: ACM Press
Full text available: pdf(1.72 MB) Additional Information: full citation, abst citings, index ter

The first interprocedural modification side-effects analysis for C (MOD(than worst-case precision on programs with general-purpose pointer usa empirical results. The analysis consists of an algorithm schema correspo MODC algorithms with two independent phases: one for determining po aliases and a subsequent one for propagating interprocedural ...

14 A simple graph-based intermediate representation
Cliff Click, Michael Paleczny
March 1995 **ACM SIGPLAN Notices , Papers from the 1995 ACM SIG on Intermediate representations,** Volume 30 Issue 3
Publisher: ACM Press
Full text available: pdf(1.39 Additional Information: full citation, abst

MB) terms

We present a graph-based intermediate representation (IR) with simple s memory-cost C++ implementation. The IR uses a directed graph with lal ordered inputs but unordered outputs. Vertices are labeled with opcodes, unlabeled. We represent the CFG and basic blocks with the same vertex Each opcode is defined by a C++ class that encapsulates opcode-specific We use inheritance to abstract common opcode behavior ...

15 Efficient subtyping tests with PQ-encoding

Yoav Zibin, Joseph Yossi Gil

October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM S conference on Object oriented programming, systems, la applications OOPSLA '01**, Volume 36 Issue 11

Publisher: ACM Press

Full text available: pdf(226.81 KB) Additional Information: full citation, abst citings, index ter

Subtyping test, i.e., determining whether one type is a subtype of anothe operation during the execution of object-oriented programs. The challen hierarchy in a small space, while simulataneously making sure that subty efficient implmentation. We present a new scheme for encoding multiple inheritance hierarchies, which, in the standardized hierarchies, reduces tl previsously published schemes. The scheme is ...

16 Logical foundations of object-oriented and frame-based languages

Michael Kifer, Georg Lausen, James Wu

July 1995 **Journal of the ACM (JACM)**, Volume 42 Issue 4

Publisher: ACM Press

Full text available: pdf(7.52 MB) Additional Information: full citation, abst citings, index ter

We propose a novel formalism, called Frame Logic (abbr., F-logic), that and declarative fashion for most of the structural aspects of object-orient languages. These features include object identity, complex objects, inher types, query methods, encapsulation, and others. In a sense, F-logic stan relationship to the object-oriented paradigm as classical predicate calcul relational programming. ...

Keywords: deductive databases, frame-based languages, logic programr inheritance, object-oriented programming, proof theory, semantics, typir

17 The design space layer: supporting early design space exploration for core-
Helvio P. Peixoto, Margarida F. Jacome, Ander Royo, Juan C. Lopez
January 1999 **Proceedings of the conference on Design, automation and**
Publisher: ACM Press
Full text available: pdf(83.19 KB) Additional Information: full citation, inde

18 A debate on language and tool support for design patterns
Craig Chambers, Bill Harrison, John Vlissides
January 2000 **Proceedings of the 27th ACM SIGPLAN-SIGACT sympc of programming languages**
Publisher: ACM Press
Full text available: pdf(2.04 MB) Additional Information: full citation, abst index terms

Design patterns have earned a place in the developer's arsenal of tools ar software development. They have proved so useful, in fact, that some ha promotion to programming language features. In turn this has rekindled over the mechanism that belong in programming languages versus those served by tools. The debate comes full circle when one contemplates coc methodological tool support for patterns. The auth ...

Keywords: design patterns, programming lnguages, software developme

19 Hierarchies and relative operators in the OLAP environment
Elaheh Pourabbas, Maurizio Rafanelli
March 2000 **ACM SIGMOD Record**, Volume 29 Issue 1
Publisher: ACM Press
Full text available: pdf(585.68 KB) Additional Information: full citation, abst terms

In the last few years, numerous proposals for modelling and querying M

Databases (MDDB) are proposed. A rigorous classification of the differe hierarchies is still an open problem. In this paper we propose and discus types of hierarchies within a single dimension of a cube. These hierarchi different levels of aggregation a single dimension. Depending on them, v characterization of some OLAP operators that refer to hierarchie ...

20 The GOLD definition language (GDL): an object oriented formal specifica multidimensional databases
Juan Trujillo, Manuel Palomar, Jaime Gómez
March 2000 **Proceedings of the 2000 ACM symposium on Applied com**
Publisher: ACM Press
Full text available: pdf(421.67 KB) Additional Information: full citation, refe index terms

Keywords: OLAP, conceptual modeling, data warehouses, multidimens object-orientation

Web Results **1** - **10** of about **129,000** for **slicing class** + **optimizing**. (**0.55** sec

Photoshop and ImageReady Training
Slicing in Photoshop and ImageReady. Understanding Slices; **Slice** Types; **Slice** Lines ... Webucator offers very affordable at-your-own-pace online **classes** on ...
www.webucator.com/WebDesign/PsWeb.cfm - 22k - Cached - Similar pages

Optimizing RAM Memory Demand
More than 95% of the **class** file data could be removed such that the resulting ... Jamaica uses one additional thread to allow time **slicing** between threads. ...
www.aicas.com/jamaica/doc/html/x1258.html - 18k - Cached - Similar pages

198:198:671 Seminar in Object-oriented Programming Languages
F. Tip, JD Choi, J. Field, G. Ramalingam, "**Slicing Class** Hierarchies", ... "The Jalapeno Dynamic **Optimizing** Compiler for Java", 1999 ACM Java Grande ...
www.cs.rutgers.edu/~ryder/oosem99/topics.html - 22k -
Cached - Similar pages

198:516
Building and **optimizing** basic blocks, recovering code from expression DAGs, ... **Class** DG for OO **slicing** References: A. Rountev, A. Milanova, BG Ryder, ...
www.cs.rutgers.edu/~ryder/516/sp06/lectures/index.html - 12k -
Cached - Similar pages

IBM Research | | jdchoi | Patents
System and method for **optimizing** computer code using a compact data flow ... Method and apparatus for **slicing class** hierarchies. US Patent 6179491. ...
domino.research.ibm.com/comm/
research_people.nsf/pages/jdchoi.patents.html - 14k - Cached - Similar pages

Macromedia Fireworks Training Course
Who Should Attend: This **class** is designed for Web designers and graphics ... Creating image maps • **Optimizing** and exporting graphics • **Slicing** images ...
www.ledet.com/outlines/html/fireworks8.shtml - 58k - Cached - Similar pages

Adobe Photoshop ImageReady Training Course
In this course you will save files in common web formats and **optimize** them for quick ... Prerequisites: Because of the amount of material we cover in **class**, ...
www.ledet.com/outlines/html/photoshopimageready.shtml - 61k - Cached - Similar pages
[More results from www.ledet.com]

CUIN 7317, The Visual Representation of Information
Optimizing Animated GIFs and Minimizing Color Palettes Available online: ... **Slicing** larger graphics and then animating sections Examples: ...
www.coe.uh.edu/courses/cuin7317/ students/**class**11/**class**11.html - 13k - Cached - Similar pages

Image **Slicing** - How and Why to **Slice** Web Images
Simply **slicing** and saving each section with the same compression settings is pointless; but by **optimizing**, you can reduce your images by several KB's and ...
graphicssoft.about.com/od/webgraphics/a/whyslice.htm - 28k - Cached - Similar pages

Training Macromedia Fireworks AZ in New York NY New York
Slice complex images into sections and use the **Optimize** panel and ... and other job related **classes** in areas like Office Administration and Bookkeeping, ...
www.training-**class**es.com/course_hierarchy/courses/7004_Macromedia_Fireworks_A-Z.php - 22k - Cached - Similar pages

Try your search again on Google Book Search

Goooooooooogle ▶

Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

slicing class + optimizing Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

Web Results **1** - **10** of about **80,500** for **unused data item hierarchical merg**

[PPT] A Unified Framework for Schedule and Storage **Optimization**
File Format: Microsoft Powerpoint - View as HTML
Consider **merging** two stages in a program ? L C S. StreamIt: **hierarchical** stream **...** At this time, **data items** are routed using a simple dimension-ordered **...**
cag.csail.mit.edu/~mgordon/raw_backend.ppt - Similar pages

[PDF] Bandwidth Skimming: A Technique for Cost-Effective Video-on-Demand
File Format: PDF/Adobe Acrobat - View as HTML
data required to effect the **merge** is received, and that each individual **item** is received in time to meet its viewing deadline. **...**
www.cs.washington.edu/homes/ zahorjan/papers/bandwidth-skimming.pdf - Similar pages

The Alsoft Collection: a review of Alsoft's power utilities
4. MenuExtend (v 1.00) This little gem makes the System 7 Apple menu **hierarchical**. When you drag down to an Apple menu **item**, if it is a folder, **...**
www.savetz.com/ku/ku/duroux_alsoft_collection.html - 16k - Cached - Similar pages

Index
hiding factory methods; **hierarchical** file system (HFS). compiler **data** sets · defining file **...** length of **data items**, finding; LENGTH OF special register **...**
publib.boulder.ibm.com/infocenter/ pdthelp/topic/com.ibm.entcobol4.doc/pgindex.htm - 377k - Cached - Similar pages

[PDF] Code and Toggle Coverage
File Format: PDF/Adobe Acrobat - View as HTML
(mind the differences related to collecting **hierarchical data** and **data** for units **...** It helps to verify the quality of the stimulus and locate **unused ...**

www.aldec.com/training/riviera/ 2006.02/pdf/C07_Code_Coverage.pdf - Similar pages

[PDF] Efficient Simulation for **Hierarchical** and Partitioned Circuits
File Format: PDF/Adobe Acrobat
of the **unused** circuits. Each time a new set of operands. is supplied to one unit, **...** contains several **data items**, including a pointer to the **...**
doi.ieeecomputersociety.org/10.1109/ICVD.1999.745154 - Similar pages

Xcell Journal Online -- Pin Assignment article Issue 55
PlanAhead software provides a complete environment to make this **hierarchical** methodology interactive and easy to use by presenting design **data** through the **...**
www.xilinx.com/publications/ xcellonline/xcell_55/xc_planahead55.htm - 34k - Cached - Similar pages

[DOC] Survivability
File Format: Microsoft Word 2000 - View as HTML
Are there **unused** or unexpected components? This provides an opportunity for **... Data items** of interest are the Web pages manipulated by GeoWorlds (both **...**
www.objs.com/DASADA/CDSA2001.doc - Similar pages

Driver Guide - free downloads - freeware and shareware Finance **...**
Optimize your expenditure based on current income **data**. **...** s unique recursive tree view is a new paradigm for organizing **hierarchical data** it is hands-down **...**
drn.digitalriver.com/category.php%5Baction%5Dbrowse& i=30&id=115&f=%7C%7C%7C%7C&s=product.... - 62k - Cached - Similar pages

Internet › HTML Tools- Free Downloads at Simtel.net
94% of computers have corrupt, **unused**, and possibly harmful files. Run a FREE registry scan to clean, repair, and **optimize** your system with the leading and **...**
www.simtel.net/category.php%5Baction%5Dbrowse& i=20&id=152&f=% 7C%7C%7C%7C&s=product.date_r... - 70k - Cached - Similar pages

Try your search again on Google Book Search

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 Next

unused data item hierarchical merge Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

Google Scholar BETA optimize unused data objects merge 1990 - 2

Scholar Results **1 - 10** of about **976** for **optimize unused data objects merge**

Iterative type analysis and extended message splitting: Optimizing dynamically-typed object-oriented ... - group of 14 »
CJ Chambers, DJ Ungar - Higher-Order and Symbolic Computation, 1991 - Springer
... the compiler is frequently able to **optimize** away the ... Standard **data** flow techniques,
on the other hand ... important for dynami- cally-typed **object**-oriented languages ...
Cited by 98 - Web Search

All articles Recent articles

[PS] The Design and Implementation of the Self Compiler, an Optimizing Compiler for Object-Oriented ... - group of 7 »
C Chambers - 1992 - research.sun.com
... 13.1.4 Interactions between Debugging and **Optimization** ... terms of implementations of
existing abstract **data** types, and ... to invoke operations on **objects** of unknown ...
Cited by 126 - View as HTML - Web Search - Library Search

Indexing and querying XML **data** for regular path expressions - group of 41 »
Q Li, B Moon - ... of the 27th International Conference on Very Large **Data** ..., 2001 - nike.psu.edu
... necessary until all the reserved spaces (ie, **unused** or- der ... traverse the hi-erarchy
of XML **objects** in either ... and bottom-up approaches for XML **data** of certain ...
Cited by 373 - View as HTML - Web Search - BL Direct

Honey, I shrunk the XQuery!: an XML algebra optimization approach - group of 3 »
X Zhang, B Pielech, EA Rundesnteiner - ... fourth international workshop on Web information and **data** ..., 2002 - portal.acm.org

... and relational **data** in the loading manager ... XAT, introduced in Section 2, for
optimization and execution ... see Figure 9). XAT Cleaner eliminates **unused** columns in ...
Cited by 26 - Web Search

INFRASTRUCTURE, QUERY OPTIMIZATION, **DATA** WAREHOUSING AND **DATA** MINING FOR SCIENTIFIC SIMULATION - group of 2 »
Y Huang - 2002 - nd.edu
... 30 4.1 **Objects**, Reactions and Processes . . . . 78 6.4 Populate The **Data** Warehouse . . . . 81 6.5 Query **Optimization** . . . .
Cited by 2 - View as HTML - Web Search - Library Search

Whole-Program Optimization of Object-Oriented Languages - group of 5 »
JA Dean - 1996 - marvin.kset.org
... away this flexibility when it is **unused** by a ... class testing allows the compiler to
optimize message sends ... **data** to determine where it is profitable to compile ...
Cited by 28 - View as HTML - Web Search - Library Search

Marmot: an optimizing compiler for Java - group of 17 »
R Fitzgerald, TB Knoblock, E Ruf, B Steensgaard, D ... - Software-Practice and Experience, 2000 - doi.wiley.com
... prior to conversion to remove **unused** methods from ... This **optimization** is deferred until
after register allocation ... the Marmot libraries and their **data** layouts are ...
Cited by 116 - Web Search - BL Direct

Data and memory optimization techniques for embedded systems
PR Panda, F Catthoor, ND Dutt, K Danckaert, E ... - ACM Transactions on Design Automation of Electronic Systems ..., 2001 - portal.acm.org
... the code using a loop- **merging** transformation, gives ... as merely geometrical **objects**,
without execution ... is particularly important to **optimize data** transfers and ...
Cited by 99 - Web Search - BL Direct

Whole-program optimization of object-oriented languages - group of 10 »

C Chambers, J Dean, D Grove - University of Washington Seattle, Technical Report 96-06, 1996 - research.ibm.com
... a closure is stored in a **data** structure that ... **Unused** potential will be optimized
automatically and simply ... When attempting to **optimize** a dynamic dispatch, we ...
Cited by 56 - View as HTML - Web Search

A Portable Approach to Dynamic Optimization in Run-time Specialization - group of 4 »
H Masuhara, A Yonezawa - New Generation Computing, 2001 - graco.cu-tokyo.ac.jp
... in Proceedings of Second Symposium on Programs as **Data Objects** (PADO-II ... Some
optimization algorithms are ... live local variables, and the types of **unused** ones do ...
Cited by 5 - View as HTML - Web Search - BL Direct

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

Slicing class hierarchies in C++

Full text Pdf (2.08 MB)

Source **Conference on Object Oriented Programming Systems Langua**
archive
Proceedings of the 11th ACM SIGPLAN conference on Object-
programming, systems, languages, and applications table of cor
San Jose, California, United States
Pages: 179 - 197
Year of Publication: 1996
ISSN:0362-1340
Also published in ...

Authors
Frank Tip IBM T.J. Watson Research Center, P.O. Box 704, Yorkt
Jong-Deok Choi IBM T.J. Watson Research Center, P.O. Box 704, Yorkt
John Field IBM T.J. Watson Research Center, P.O. Box 704, Yorkt
G. Ramalingam IBM T.J. Watson Research Center, P.O. Box 704, Yorkt

Sponsor SIGPLAN: ACM Special Interest Group on Programming Langua
Publisher ACM Press New York, NY, USA

↑ ABSTRACT

This paper describes an algorithm for *slicing* class hierarchies in C++ progran
hierarchy (a collection of C++ classes and inheritance relations among them)

uses the hierarchy, the algorithm eliminates from the hierarchy those data me functions, classes, and inheritance relations that are unnecessary for ensuring is maintained.Class slicing is especially useful when the program *P* is generat program *P'* by a *statement slicing* algorithm. Such an algorithm eliminates sta irrelevant to a set of slicing *criteria*---program points of particular interest. Th considerable previous work on statement slicing, and it will not be the concer However, the combination of statement slicing and class slicing for C++ has t applications: First, class slicing can enhance statement slicing's utility in prog understanding applications, by eliminating both executable *and* declarative pr irrelevant to the slicing criteria. Second, the combination of the two slicing al decrease the space requirements of programs that do not use all the componer Such a situation is particularly common in programs that use class libraries.

↑ REFERENCES

Note: OCR errors may be found in this Reference List extracted from the full opted to expose the complete List rather than only correct and linked referenc

1 ACCREDITED STANDARDS COMMITTEE X3, I. P. S. Working paper international standard for information systems---programming language C++. 1995.

2 AGESEN, O. Constraint-based type inference and parametric polymorphis First International Static Analysis Symposium (SAS'94) (September 1994), 7 LNCS vol. 864.

3 Ole Agesen, Concrete type inference: delivering object-oriented applicatio University, Stanford, CA, 1996

4 Ole Agesen , David Ungar, Sifting out the gold: delivering compact applic exploratory object-oriented programming environment, ACM SIGPLAN Not 370, Oct. 1994

5 Hiralal Agrawal, On slicing programs with jump statements, Proceedings 1994 conference on Programming language design and implementation, p.30 1994, Orlando, Florida, United States

6 Hiralal Agrawal , Richard A. DeMillo , Eugene H. Spafford, Dynamic slic unconstrained pointers, Proceedings of the symposium on Testing, analysis, a

73, October 08-10, 1991, Victoria, British Columbia, Canada

7 David F. Bacon , Peter F. Sweeney, Fast static analysis of C++ virtual func of the 11th ACM SIGPLAN conference on Object-oriented programming, sys applications, p.324-341, October 06-10, 1996, San Jose, California, United St

8 BACON, D. F., WEGMAN, M., AND ZADECK, F. K. Rapid type inferer RC 1234, IBM Thomas J. Watson Research Center, 1995.

9 Thomas Ball , Susan Horwitz, Slicing Programs with Arbitrary Control-flc First International Workshop on Automated and Algorithmic Debugging, p.2(1993

10 Brad Calder , Dirk Grunwald, Reducing indirect function call overhead ir Proceedings of the 21st ACM SIGPLAN-SIGACT symposium on Principles (languages, p.397-408, January 16-19, 1994, Portland, Oregon, United States

11 CARINI, P. R., HIND, M., AND SRINIVASAN, H. Flowsensitive type a Rep. RC 20267, IBM T.J. Watson Research Center, 1995.

12 Jong-Deok Choi , Jeanne Ferrante, Static slicing in the presence of goto s Transactions on Programming Languages and Systems (TOPLAS), v.16 n.4, J

13 DEAN, j., AND CHAMBERS, C. Optimization of objectoriented prograı hierarchy analysis. Tech. Rep. 94-12-01, Department of Computer Science, U Washington at Seattle, December 1994.

14 Margaret A. Ellis , Bjarne Stroustrup, The annotated C++ reference manı Longman Publishing Co., Inc., Boston, MA, 1990

15 ERNST, M. Practical fine-grained static slicing of optimized code. Tech. Microsoft Research, Redmond, WA, 1994.

16 Mary F. Fernández, Simple and effective link-time optimization of Modu Proceedings of the ACM SIGPLAN 1995 conference on Programming langua implementation, p.103-115, June 18-21, 1995, La Jolla, California, United Sta

17 John Field , G. Ramalingam , Frank Tip, Parametric program slicing, Pro ACM SIGPLAN-SIGACT symposium on Principles of programming languag 23-25, 1995, San Francisco, California, United States

18 Susan Horwitz , Thomas Reps , David Binkley, Interprocedural slicing us graphs, ACM Transactions on Programming Languages and Systems (TOPL/ Jan. 1990

19 KRISHNASWAM~, A. Program slicing: An application of object-oriente graphs. Technical report TR94-108, Dept. of Computer Science, Clemson Un

20 Loren Larsen , Mary Jean Harrold, Slicing object-oriented software, Proc international conference on Software engineering, p.495-505, March 25-29, 1

21 LIVADAS, P. E., AND CROLL, S. Program slicing. Report serc-tr-6 l-f, Department, University of Florida, 1992.

22 Jens Palsberg , Michael I. Schwartzbach, Object-oriented type inference, Notices, v.26 n.11, p.146-161, Nov. 1991

23 PANDE, H. D., AND RYOER, B. G. Static type determination and aliasi LCSR-TR-250-A, Rutgers University, October 1995.

24 John Plevyak , Andrew A. Chien, Precise concrete type inference for obje ACM SIGPLAN Notices, v.29 n.10, p.324-340, Oct. 1994

25 Thomas Reps , Susan Horwitz , Mooly Sagiv , Genevieve Rosay, Speedi Proceedings of the 2nd ACM SIGSOFT symposium on Foundations of softwa 20, December 06-09, 1994, New Orleans, Louisiana, United States

26 Jonathan G. Rossie, Jr. , Daniel P. Friedman, An algebraic semantics of s SIGPLAN Notices, v.30 n.10, p.187-199, Oct. 17, 1995

27 SAKKINEN, M. A critique of the inheritance principles of C++. Comput (1992), 69--I I0.

28 Amitabh Srivastava, Unreachable procedures in object-oriented programi Programming Languages and Systems (LOPLAS), v.1 n.4, p.355-364, Dec. 1

29 TIP, F. A survey of program slicing techniques. Journal of Programming 121-189.

30 Mark David Weiser, Program slices: formal, psychological, and practical automatic program abstraction method, 1979

↑ CITINGS *17*

Bi-Xin Li , Xiao-Cong Fan , Jun Pang , Jian-Jun Zhao, A model for slicing JA hierarchically, Journal of Computer Science and Technology, v.19 n.6, p.848-

G. Ramalingam , Harini Srinivasan, A member lookup algorithm for C++, AC v.32 n.5, p.18-30, May 1997

Amie L. Souter , Lori L. Pollock , Dixie Hisley, Inter-class def-use analysis w representations, ACM SIGSOFT Software Engineering Notes, v.24 n.5, p.47-

Derek Rayside , Kostas Kontogiannis, On the syllogistic structure of object-o Proceedings of the 23rd international conference on Software engineering, p. 2001, Toronto, Ontario, Canada

Frank Tip , Chris Laffra , Peter F. Sweeney , David Streeter, Practical experie extractor for Java, ACM SIGPLAN Notices, v.34 n.10, p.292-305, Oct. 1999

Peter F. Sweeney , Frank Tip, A study of dead data members in C++ applicati Notices, v.33 n.5, p.324-332, May 1998

David F. Bacon , Peter F. Sweeney, Fast static analysis of C++ virtual functio SIGPLAN Notices, v.31 n.10, p.324-341, Oct. 1996

Frank Tip , Peter F. Sweeney, Class hierarchy specialization, ACM SIGPLAN p.271-285, Oct. 1997

Martin P. Robillard , Gail C. Murphy, Concern graphs: finding and describing structural program dependencies, Proceedings of the 24th international confer engineering, May 19-25, 2002, Orlando, Florida

Gregor Snelting , Frank Tip, Reengineering class hierarchies using concept an SIGSOFT Software Engineering Notes, v.23 n.6, p.99-110, Nov. 1998

Katsuhisa Maruyama , Ken-Ichi Shima, An Automatic Class Generation Mec Method Integration, IEEE Transactions on Software Engineering, v.26 n.5, p.

Derek Rayside , Kostas Kontogiannis, Extracting Java library subsets for depl systems, Science of Computer Programming, v.45 n.2-3, p.245-270, Novemb

Gregor Snelting , Frank Tip, Understanding class hierarchies using concept ai Transactions on Programming Languages and Systems (TOPLAS), v.22 n.3,]

Frank Tip , Peter F. Sweeney , Chris Laffra , Aldo Eisma , David Streeter, Pra techniques for Java, ACM Transactions on Programming Languages and Syst n.6, p.625-666, November 2002

Ramkrishna Chatterjee , Barbara G. Ryder , William A. Landi, Complexity of Java in the Presence of Exceptions, IEEE Transactions on Software Engineeri 512, June 2001

Barbara G. Ryder , William A. Landi , Philip A. Stocks , Sean Zhang , Rita A interprocedural modification side-effect analysis with pointer aliasing, ACM ' Programming Languages and Systems (TOPLAS), v.23 n.2, p.105-186, Marc

Baowen Xu , Ju Qian , Xiaofang Zhang , Zhongqiang Wu , Lin Chen, A brief slicing, ACM SIGSOFT Software Engineering Notes, v.30 n.2, March 2005

↑ INDEX TERMS

Primary Classification:

D. Software
↳ **D.1** PROGRAMMING TECHNIQUES

Additional Classification:

D. Software
↳ **D.3** PROGRAMMING LANGUAGES
↳ **D.3.2** Language Classifications

↳ **Nouns:** C++
↳ **D.3.3** Language Constructs and Features
↳ **Subjects:** Data types and structures

F. Theory of Computation
↳ **F.3** LOGICS AND MEANINGS OF PROGRAMS
↳ **F.3.3** Studies of Program Constructs
↳ **Subjects:** Type structure

General Terms:
Algorithms, Languages, Performance, Theory

Collaborative Colleagues:

Jong-Deok Choi:	Bowen Alpern	Jeanne Ferrante	Sam Midkiff
	Koenraad De Bosschere	John Field	Samuel P. Midkiff
	Michael Burke	Stephen Fink	B. P. Miller
	Michael G. Burke	David Grove	Barton P. Mill
	Paul Carini	Manish Gupta	Sang L. Min
	Paul R. Carini	Michael Hind	Sang Lyul Mi
	Soumen Chakrabarti	Clinton Jeffery	Robert H. B. Netzer
	Mark Charney	William G. Landi	Ton Ngo
	Mark Christiaens	Keunwoo Lee	Robert O'Callahan
	Anthony Cocchi	Raimondas Lencevicius	G. Ramalinga
	R. Cytron	Derek Lieber	Michiel Ronss
	Ron Cytron	Alexey Loginov	Barbara G. Ry
	J. Ferrante	Thomas J. Marlowe	Vivek Sarkar
John Field:	Jan A. Bergstra	Tim Teitelbaum	
	Jong-Deok Choi	Frank Tip	
	T. B. Dinesh	Carlos A. Varela	
	Deepak Goyal	Alex Warshavsky	
	Jan Heering		
	Roman Manevich		
	G. Ramalingam		
	Mooly Sagiv		
	Bruce Segee		
	Gregor Snelting		
G. Ramalingam:	Jong-Deok Choi	Thomas Reps	
	John Field	Thomas W. Reps	
	G. Goos	Mooly Sagiv	
	Deepak Goyal	Junehwa Song	
	J. Hartmanis	Harini Srinivasan	
	Michelle Y. Kim	Frank Tip	

	J. van Leeuwen	Alex Warshavsky	
	Roman Manevich	Eran Yahav	
	Raymond Miller	Byoung-Kee Yi	
	C. Pandu Rangan		
Frank Tip:	Dirk Bäumer	Robert Fuhrer	Fenil Shah
	Ittai Balaban	Markus Keller	Gregor Sneltir
	Thomas Ball	Adam Kieżun	Maximilian
	A. Michael Berman	Adam Kiezun	Stoerzer
	Ophelia Chesley	Paul Klint	David Streeter
	Jong-Deok Choi	Chris Laffra	Peter F. Sweer
	Arie Deursen	Jens Palsberg	
	T. B. Dinesh	G. Ramalingam	
	Aldo Eisma	Xiaoxia Ren	
	John Field	Barbara G. Ryder	

↑ **Peer to Peer - Readers of this Article have also read:**

- Data structures for quadtree approximation and compression **Communica**
  28, 9
  Hanan Samet

- A hierarchical single-key-lock access control using the Chinese remainder
  of the 1992 ACM/SIGAPP Symposium on Applied computing
  Kim S. Lee , Huizhu Lu , D. D. Fisher

- The GemStone object database management system **Communications of**
  Paul Butterworth , Allen Otis , Jacob Stein

- An intelligent component database for behavioral synthesis **Proceedings c**
  ACM/IEEE conference on Design automation
  Gwo-Dong Chen , Daniel D. Gajski

- Putting innovation to work: adoption strategies for multimedia communica
  Communications of the ACM 34, 12
  Ellen Francik , Susan Ehrlich Rudman , Donna Cooper , Stephen Levine

↑ **This Article has also been published in:**

- **ACM SIGPLAN Notices**
  Volume 31 , Issue 10 Oct. 1996